PATENT
B208-685

EXPRESS MAIL LABEL NO. EL030870561US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants	:	Fumiaki Takahashi, et al.
Serial No.	:	Unassigned (Parent 08/942,729 filed 10/2/97)
For	:	SYSTEM CONTROL METHOD AND SYSTEM CONTROL APPARATUS
Filed	:	January 29, 1999
Examiner	:	Unassigned (Parent - L. Luu)
Art Unit	:	Unassigned (Parent - 2756)

Assistant Commissioner
for Patents
Washington, D.C. 20231

Sir:

CLAIM FOR PRIORITY

Claim is made under 35 USC § 119 for the benefit of the filing date of Japanese Patent Application No. Hei 05-189633 (filed 7/30/93); Hei 05-189634 (filed 7/30/93); Hei 05-189636 (filed 7/30/93) and Hei 05-189637 (filed 7/30/93); copies of which have been filed in prior application Serial No. 08/281,355.

Respectfully submitted,

John J. Torrente

ROBIN, BLECKER & DALEY
330 Madison Avenue
New York, New York 10017
(212) 682-9640
January 29, 1999

John J. Torrente
Registration No. 26,359
An Attorney of Record

24.04.95

PATENT OFFICE

JAPANESE GOVERNMENT

This is to certify that the annexed is a true copy of the following application as filed with this office.

Date of Application: July 30, 1993

Application Number: Patent Application No. Hei 05-189633

Applicant(s): Canon Kabushiki Kaisha

August 26, 1994

Commisioner,

Patent Office Akira Takashima

Issuance No.Hei 06-3043518

(Name of Document) Application for Patent

(File Number) 2589043

(Date of Submission) July 30, 1993

(Addressee) To Commissioner, Patnet Office,
Wataru Asoh

(International Patent Classification) H04N 5/232

(Title of Invention) SYSTEM CONTROL METHOD AND APPARATUS USING THE SAME

(Number of Claims) 16

(Inventor)
(Address) c/o Canon Kabushiki Kaisha, 3-30-2, Shimomaruko, Ohta-ku, Tokyo
(Name) Hideaki Kawamura

(Inventor)
(Address) c/o Canon Kabushiki Kaisha, 3-30-2, Shimomaruko, Ohta-ku, Tokyo
(Name) Akira Suga

(Inventor)
(Address) c/o Canon Kabushiki Kaisha, 3-30-2, Shimomaruko, Ohta-ku, Tokyo
(Name) Akira Mamiya

(Inventor)
(Address) c/o Canon Kabushiki Kaisha, 3-30-2, Shimomaruko, Ohta-ku, Tokyo
(Name) Fumiaki Takahashi

(Inventor)
(Address) c/o Canon Kabushiki Kaisha, 3-30-2, Shimomaruko, Ohta-ku, Tokyo
(Name) Kenji Hatori

(Inventor)
(Address) c/o Canon Kabushiki Kaisha, 3-30-2, Shimomaruko, Ohta-ku, Tokyo
(Name) Takashi Aizawa

(Applicant for Patent)
(Identification Number) 000001007
(Postal Code) 146
(Address) 3-30-2, Shimomaruko, Ohta-ku, Tokyo
(Name) Canon Kabushiki Kaisha
(Representative) Hajime Mitarai
(Telephone Number) 03-3758-2111

(Agent)
(Identification Number) 100069877

(Postal Code) 146
(Address) c/o Canon Kabushiki Kaisha, 3-30-2, Shimomaruko, Ohta-ku, Tokyo
(Patent Attorney)
(Name) Giichi Marushima
(Telephone Number) 03-3758-2111

(Indication of Fee)
(Method of Payment) Deposit
(Deposit Account Number) 011224
(Amount of Payment) 21000

(List of Submitted Articles)
(Name of Article) Specification 1
(Name of Article) Drawings 1
(Name of Article) Written Abstract 1
(General Power of Attorney Number) 9003707

(Necessity of Proof) Necessary

(Name of Document)

Specification

(Title of the Invention)

SYSTEM CONTROL METHOD AND APPARATUS USING THE SAME

(Scope of Claim for Patent)

(Claim 1)

A control apparatus comprising a plurality of object embedded peripheral devices and a controller for unitarily controlling said plurality of peripheral devices, as said plurality of peripheral devices are connected via a common communication line to said controller, characterized in that said controller is responsive to connection of each of said peripheral devices thereto for reading the control information, as stored in the connected one of said peripheral device, over said communication line to a predetermined memory area in said controller, thereby making it possible for said controller to control by itself said peripheral devices and that the given commands by said controller in itself are transmitted in the form of signals over said communication line to each of said connected peripheral devices.

(Claim 2)

A system control method for an apparatus, said apparatus having a plurality of object embedded peripheral devices, one and the same controller for unitarily controlling said peripheral devices and a common duplex interface for connecting said controller to said peripheral devices either to control said peripheral devices or to coordinate inputting and outputting of data, wherein said common duplex interface is made use of to enable said controller to control each of said peripheral devices.

(Claim 3)

A system control method according to claim 2, wherein said object embedded peripheral devices and said controller are provided with means for receiving and transmitting messages (control commands and data input/output commands) from and to one another.

(Claim 4)

A system control method according to claim 3, wherein said object embedded peripheral devices and said controller have groups of methods (sequence of instructions to execute, functions, subroutines) to be selected by said messages, whereby environment is made such that the selected methods are activated.

(Claim 5)

A system control method according to claim 4, wherein said object embedded peripheral devices and said controller are encapsulated so that, as their internal data such as those about the internal states thereof and those of the variable parameters are hidden inside said internal data are accessed indirectly by calling on said methods.

(Claim 6)

A system control method according to claim 5, wherein said object embedded peripheral devices have their control panels and display devices in the form of graphical user interfaces (GUIs) so that they operate themselves and present their own displays by using external controllers, whereby means is provided for sending that GUI to said external controllers.

(Claim 7)

A system control method according to claim 6, wherein said object embedded peripheral devices have tables of methods defined in correspondence to the modes of operation of the control panels of said graphical user interfaces (GUIs), whereby means is provided for sending that table to said external controllers.

(Claim 8)

A system control method according to claim 7, wherein said object embedded peripheral devices have their

graphical user interfaces (GUIs) and the method definition tables conglomerated in the form of GUI objects, whereby means is provided for sending this GUI object to the external controller.

(Claim 9)

A system control method according to claim 7, wherein said object embedded peripheral devices have their graphical user interfaces (GUIs) and method definition tables realized in common by the language in which the GUIs are described, whereby means is for sending this GUI description language to the external controller.

(Claim 10)

A system control method according to claim 2, wherein arrangement is made such that said controller becomes to have display means for the state of physical connection of the plurality of object embedded peripheral devices thereto.

(Claim 11)

A system control method according to claim 2, wherein arrangement is made such that said controller becomes to have display means for the status of the plurality of object embedded peripheral devices and altering means therefor.

(Claim 12)

A system control method according to claim 12, wherein arrangement is made such that said controller becomes to have means for reading the graphical user interfaces and the method definition tables from the plurality of object embedded peripheral devices thereto and display means therefor.

(Claim 13)

A system control method according to claim 12, wherein arrangement is made such that said controller becomes to have the functions which are realized by the GUI objects.

(Claim 14)

A system control method according to claim 12, wherein arrangement is made such that said controller becomes to have the functions which are realized by the GUI description language.

(Claim 15)

A system control method according to claim 12, wherein said controller is provided with means for retrieving the corresponding method to the actuation of a button or the like of the object embedded peripheral device from the definition table and means for sending it as a message to the object (peripheral device) aimed at.

(Claim 16)

A system control method according to claim 2, wherein arrangement is made such that said controller becomes to have the function of downloading programs and data as taken from an external peripheral device for the purpose either of upping its own functions or of remedying the damaged functions by bugs or the like.

(Detailed Description of the Invention)

(0001)

(Field of Utility on Industry)

This invention is suited to be used in controlling the system of multimedia devices that handle texts, sound, still pictures, motion pictures and like data.

(0002)

(Prior Art)

In the field of art of audio, video, TV and like AV instruments which have so far employed analog techniques at the center, digitization is rapidly advancing in recent years. Along with the widespread use of digital techniques in the data of texts and still pictures have come the trend of unitarily handling so-called multimedia comprised of the data of texts, sound, still pictures and motion pictures in the interior of a computer.

(0003)

(Subjects the invention is to Solve)

However, up to now, in the case when to utilize the multimedia devices (digital camera, CD-ROM player, scanner, sound board, video board and other sound input/output devices and other video input/output devices) in operative connection to the computer, it has been the common practice to install into the computer an item of software, which is dedicated solely to driving a particular one of these devices, called the "application" or "device driver."

(0004)

On advent of every new multimedia device, therefore, relying on this customary method leads to the necessity of preparing an corresponding number of different new applications or device drivers to the number of combinations of computers and OS's (Operating Systems). The development of many such items of software cost huge labor and long time. Another problem was that the high efficiency and high speed of control were impossible.

(0005)

With the use of this method, it is also impossible in the general case that another computer, though connected to the LAN, transparently uses that multimedia device. Hence the concept that every computer is able to access every peripheral device over the LAN, say a multimedia system, could not be realized.

(0006)

(Means for Solving the Subjects)

To solve the problems described above, according

to the invention, a control apparatus is constructed, comprising a plurality of object embedded peripheral devices and a controller arranged on connection with the aforesaid peripheral devices via a common communication line to unitarily control all the aforesaid peripheral devices, wherein, as the peripheral devices store control information therefor in themselves, the controller is responsive to connection of any one of the peripheral devices thereto for reading the control information from the connected one of the peripheral devices via the communication line to a predetermined memory area in the interior of the controller and for writing it therein to a predetermined format, thereby making it possible for the controller to control by itself the peripheral device, and wherein, as the controller constructs commands in itself, their signals are transmitted over the same communication line as described above to any one of the aforesaid peripheral devices.

(0007)

To solve the same problems, according to the invention as applied to a system comprising a plurality of object embedded peripheral devices, one controller for unitarily controlling all the peripheral devices and a common duplex interface connecting the controller to the plurality of peripheral devices either to control the peripheral devices or to coordinate inputting and outputting of data, use is made of a system control method that it is through the aforesaid common duplex interface

that each of the aforesaid peripheral devices is made possible to control by the aforesaid controller.

(0008)

(Function)

Hence the control of the multimedia device can be made without having to develop any special part of software such as "application" or "device driver" mentioned before. Moreover, it is made possible to realize an environment that permits another controller to utilize the multimedia device transparently in common over the LAN.

(0009)

(Embodiments)

The present invention is next described in detail in connection with embodiments thereof by reference to the drawings.

(0010)

It should first be noted that in the invention the multimedia devices are individually taken as objects and a system control technique is adopted that the controller manages all these objects unitarily.

(0011)

Each of the objects is made manageable by the controller. For this purpose, the objects have the function that they send their own functions and/or control means out to the controller. This leads to obviate the necessity of previously installing the control program in the controller itself, as was heretofore usual. Thus,

all what is necessary to realize establishment of the control system is only connect them to the controller.

(0012)

Another feature is that, as the aforesaid control means is sent from the connected one of the objects to the controller, the controller is provided with means for displaying the sent control means and for permitting one who actually makes control commands to carry out actuation. By this it is made possible for the controller to be as a center in charge of all the multimedia devices. Even for a new multimedia device to come, the present state of the apparatus suffices for coping with it without having to undertake the preparation of any application. Thus, achievement of great increases of the flexibility and versatility can be realized.

(0013)

Incidentally, while the invention adopts the object orientation, this concept itself is fully explained in the documents, for example, Ishizuka: "Object-Oriented Programming" ASCII Publishing 1988; Sakai: "Introduction to Object Oriented Technique" Ohm Co. 1990; and B.J. Cocks: "Object Oriented Programming" Toppan 1988. In the following, therefore, the fundamental technology is no more explained in describing embodiments of the invention.

(0014)

On this object orientation, the spotlight of attention is being focused from the standpoint of the

recent trend of improving the efficiency of program development environment or as such. Moreover this idea can be widely applied even to the OS's and multimedia database. In particular, the characteristic perception to the object orientation is:

(1) Encapsulation;

(2) Inheritance; and

(3) Messaging

With these three points as a base, the present invention has made attempts to develop and expand the technology so that it can be applied to the control of multimedia devices.

(0015)

Fig. 1 shows a logical scheme of connection of a multimedia controller with multimedia devices employing the concept of object orientation of the invention. With the multimedia controller 1 at the core, an equal number of communication paths to the number multimedia devices 2 are established so that, when transmitting various items of information, every one of the multimedia device 2 can make direct conversation in one-to-one basis with the multimedia controller 1. It will be appreciated that the control is made by a method of transmitting messages to each other over that communication path. The term "multimedia devices" herein used, specifically speaking, means all of those devices which handle multimedia data, namely, CD players, digital VTRs, digital cameras, digital TV sets and other AV

devices and digital FAX, digital copiers, printers and other OA devices.

(0016)

The controller is assumed here to be part of hardware dedicated solely to this purpose. But it is also possible to realize an equivalent controller by installing an especial OS and a particular application on the commonly available processor in the personal computer or word processor.

(0017)

Next, in Fig. 2, there are shown three configurations (a) to (c) for physically connecting a multimedia controller to a number of multimedia devices to establish the respective duplex communication paths.

(0018)

The daisy chain line of Fig. 2(a) is employed in SCSI bus (ANSI X3.131-1986). The star configuration of Fig. 2(b) is employed in Ethernet (IEEE 802.3) 10BaseT. The multipoint line of Fig. 2(c) is employed in Ethernet 10Base2/5.

(0019)

It is also to be noted that with regard to another possible configuration, there is GPIB (IEEE 4888) as obtained by mixing the (a) to (c). Even in Ethernet, the (b) and (c) may be mixed. With regard to another possible communication systems, there are optical fiber cables and ISDN. So it is to be understood that, besides those of Fig. 2, many other combinations are possible

to make and may be selectively employed as desired.

(0020)

How to establish such duplex communication paths and which to select are not essential to the invention, so no particular remarks are given except that, as the communication system differs from one to another, some physical limitations are laid on the transfer speed, the number of connected devices, the length of the cable, the shape of the connector, etc. For the each other's transmission of messages, because the protocols have their hierarchies differentiated from each other, these limitations are out of question. In order to insure that the peripheral devices each are connected to the controller reliably and accurately, however, there is need to provide for the system with at least one physically (mechanically and/or electrically) common interface.

(0021)

To realize high speed transmission of data such as those of motion pictures, it is recommended to employ what is faster than Ethernet, that is, the optical transmission such as FDDI (Fiber Distributed Data Interface) or B-ISDN. But, in here, for the purpose of simplicity of explanation, discussions are conducted on assumption that Ethernet 10Base2(/T) which, because of its cheap price, is widely used is adopted as the common communication connector.

(0022)

Next, the internal pieces of hardware of the usual multimedia device are shown in an block diagram of Fig. 3.

(0023)

A plurality of multimedia devices are connected via a LAN 4 to the controller. Now this LAN is Ethernet so each of these devices is provided with a transmission interface 20 for the protocol (TCP/IP). This can be realized by using an exclusive LSI or the like. In here, the transmitted message itself is taken out. Conversely, a message is sent out to the controller. An example of these messages in the Objective-C is given in the general format by the following expression:

(0024)

(Target_Object Method_Name: Parameter)

In other languages, the expression takes different styles, but is similar in the following basic components:

(0025)

(1) Address of a terminal object;

(2) Selection of a method (instruction to execute); and

(3) Use of data in parameters, if any.

How to deal with this message is described in connection with the flow of software of Fig. 3.

(0026)

In the interior of the multimedia device, a CPU 11 controls an internal bus when to process all software and when to operate all hardware. The programs, the initial values and distinct database are stored in a ROM 12. A RAM 13 temporarily stores some of the data and the internal parameters representing the state of the device. When executing the programs, this RAM 13 is used as a work area. A data I/O 14 is used in accessing an internal or external medium 15 storing multimedia data. A driver 16 controls a mechanical system including an electric motor 17 and other mechanisms. Another driver 18 controls an electrical system including switches SW and LEDs or like indicators. As the multimedia data are in digital form ranging from pictures to sound to texts, the possible types the medium 15 takes are the optical disks such as CD-ROM and MD, the magnetic tapes such as DCC and DAT, and the semiconductor memory cards.

(0027)

Next, the internal pieces of hardware of the multimedia controller 1 are shown in a block diagram of Fig. 4. The controller 1 is connected via the LAN 4 to the multimedia devices. Now this LAN 4 is Ethernet, so there is an interface 31 for its communication the protocol (TCP/IP). This can be realized by utilizing an exclusive LSI or the like. In here, the transmitted message itself is taken out. Conversely messages are sent out from here to the multimedia devices.

(0028)

In the interior of the multimedia controller 1, a CPU 21 controls a bus 30 when to process all software and when to operate all hardware. The programs, the initial values and the especial database are stored in a ROM 22. Some of the data and the internal parameters representing the state of the controller are stored in a RAM 23. When executing the programs, the RAM 23 is used as a work area. With the use of a multimedia filing device 25, regardless of whether the medium is of the internal or external type, the multimedia data are recorded, retrieved, reproduced, or edited. For this purpose, accessing is controlled by a data I/O 24. A driver 28 controls an electrical system 29 including switches SW and LEDs or like indicators. A display controller 26 is connected to a display 27 constituting a man-machine interface. There is further included a mouse or like pointing device, though not shown.

(0029)

Fig. 6 is a diagram of the hierarchy of systems of software of the multimedia device. The internal parts shown in the block diagram of Fig. 3 lie at the bottom 57 labeled "hardware". An OS 58 is fundamentally in charge of this hardware. What type of OS to select is not itself particularly limited, but it is desired that the real-time facility and the multitasking capabilities that run more than one program in parallel at a time are available in combination. On this OS the software has a class

library 59 which differs with different multimedia devices for the purpose of realizing embedding of an object into that multimedia device.

(0030)

Though not shown, there is another library concerning the self-contained control panel and the control program, both of which are necessary to be controlled from the controller. On connection to the controller, this library is transmitted thereto, thus establishing the specific control of the multimedia device from the controller. To operate the timer and perform arithmetic operations, there is the C function 60 too.

(0031)

At the top of the hierarchy, there is an application 61 that controls the main system of the multimedia device, participates in communication with the multimedia controller, and acts as the user interface. By this application, the main system of the multimedia device is taken as one object so that the various controls and functions can be carried out from the controller, as the multimedia device to which messages are to be transmitted is addressed by using that object. The internal parameters, too, are taken as instance variables so that reading and altering can be carried out.

(0032)

Fig. 5 is a diagram of the hierarchy of systems of software of the multimedia controller. Its internal parts shown in the block diagram of Fig. 4 lies at the

bottom 50 labeled "hardware." An OS 51 is fundamentally in charge of this hardware. Even in here, What type of OS to select is itself not particularly limited. But it is desired that the real-time facility and the multitasking capabilities are available in combination.

(0033)

On this OS there is a window server 52 that displays the control panels of the physically connected multimedia devices and the states of the operatively connected ones of these devices over the entire system, controls their operations and coordination of inputting and outputting of data, and performs all the other functions of a GUI (Graphical User Interface). A common class library 53 stores what is beforehand ready in the controller by itself, that is, a set of basic and common components (in the form of objects) concerning the user interface, namely, buttons, slide volumes and text presentation areas and also concerning the control.

(0034)

A specific class library 55 is, on the other hand, assigned to sets of distinct components concerning the control panels and the control modes (in the form of objects) of the connected multimedia devices. This specific library, as described before, increases its content each time one more multimedia device is brought into connection with the system, as an additional class is sent from that device. This procedure will be described later in more detail. There is also a C

function 54 for the timer and arithmetic computation. At the top of the hierarchy, there is an application 56 that undertakes in controlling coordination of all the connected multimedia devices and communicating with these multimedia devices and that functions as the user interface.

(0035)

The flow of control signals and the transmission of messages between this controller and the multimedia device are described below.

(0036)

Fig. 7 shows a state of the system before the multimedia device is connected to the multimedia controller. In Fig. 7, digital data are transmitted over a communication line or LAN 4. A multimedia controller 1 controls the operation of the entirety of the system. A multimedia device 2 to be connected to the LAN 4 is shown as having a typical structure of construction. The one of software objects (hereinafter abbreviated to "objects") which always resides in the multimedia controller 1 and executes the management functions of the entirety of the system is a system director object 205.

(0037)

To embed a certain object 1064 into this connected multimedia device 2, its identification must be made distinct from the other objects on the LAN 4 so that this object functions as the multimedia device. Such

a multimedia device object is comprised of three constituent objects 1065, 1066 and 1067.

(0038)

The object 1065 named "controller for multimedia device" is in charge of hardware to realize a majority of functions of the multimedia device 2. The object 1066 named "data input to multimedia device" partakes in entering the digital data as transmitted from the other devices over the LAN 4. The object 1067 named "data output from multimedia device" partakes in transmitting the digital data to the other devices over the LAN 4.

(0039)

When the multimedia device 2 is connected to the multimedia controller 1 via the LAN 4, an object that stands in the place of that multimedia device 2 must be formed in the multimedia controller 1. To describe this deputy multimedia device object, its specification is defined in a file 1061. This file for describing the deputy multimedia device object is constructed from two parts, of which the section 1062 describes the specification of the control panel for the multimedia device 2, and the other of which the section 1063 describes the specification of an object that stands in the place of the input or output of data to or from the multimedia device. In particular, the section 1062 for describing the "control panel for the multimedia device" object realizes the function of describing the control panel so that one interacts with the multimedia device 2

by means of a GUI, that is, the function of the language in which to describe the GUI.

(0040)

Fig. 8 is a diagram to explain the state of the system after the multimedia device 2 is connected to the LAN 4. In Fig. 8, what is now formed in the multimedia controller 1 is an object 1068. In the interior of the multimedia controller 1, this object acts as substitute for that multimedia device 2, so it is called "deputy" multimedia device. This object 1068 is comprised of an object 1069 which functions as the control panel for the multimedia device 2, named "control panel for multimedia device", another object 1070 which, when to input data, functions as substitute for the object 1066 named "data input to multimedia device" so the object 1070 is called "deputy" data input to multimedia device, and still another object 1071 which functions also as substitute for the "data output from multimedia device" object 1067. This object 1071 is also called "deputy" data output from multimedia device.

(0041)

Fig. 9 is a diagram of the structure of construction of a general class library. For the objects having a similar feature, their common attribute and function have to be defined. To form an object that execute this function, a class, say a first class 1079, functions as a template. The first class 1079 to the p-th class 1085, totaling p classes, are summed up in

a library 1086. This is conventionally called "class" library. All the objects belong to the respective specified classes. The classification of the objects is defined by part 1080 of every class in respect to the data types and names of the internal variables the objects possess and the data types and names of the internal functions (usually called "class methods"), which embody means for processing data. To allow one to access to the class methods, all the codes of the class methods are tabulated relative to a point, thus forming a class method table 1081. Assuming that the number of class methods is k, a code space 1082 contains the first function code 1083 through the k-th function code 1084.

(0042)

Fig. 10 is a diagram of the structure of construction of a typical object. In Fig. 10, an object 234 comprises a portion 244 for accommodating the pointer that is to go to the class method table, communicating means 245 for messages, processing and retrieving means 246, a portion 239 for methods and a portion 235 for internal data. The "method" portion 239 comprises beginning with first data processing means 240, second data processing means 241 and so on and terminating at m-th data processing means 242, totaling m data processing means. The portion 235 for internal data comprises beginning with a first internal data 236, a second internal data 237 and so on and terminating at an n-th internal data 238, totaling n internal data.

(0043)

All the internal data in the portion 235 differ with the different objects and, therefore, are left as they stand in the interior of each of the objects. The data processing means in the "method" portion 239, on the other hand, can be used in common by the other objects, if of the same class. To assure this, therefore, a class method table 243 is provided so that the first data processing means 240 through the m-th data processing means 242 are made manageable in each of the classes. Thus, such common data processing means are shared by a number of objects which belong to the same class. To look up the class method table 243 from every object, the pointer is brought into this table from an accommodating portion 244 therefor.

(0044)

The message communicating means 245, when in receiving a message from another object, transfers it to the processing and retrieving means 246, where the message is analyzed to identify its address section and the corresponding one of the data processing means to it is retrieved from the "method" portion 239 (virtually the class method table 243). As the selected data processing means operates, the data section of the message, the internal data from the portion 235 and some external data are processed in a predetermined way. In some case, such processing will result in production of a message. If so, this message is transmitted from the

communicating means 245 to that other object.

(0045)

Fig. 11 is a diagram of the structure of construction of the system director object 205. A space 1072 accommodates a pointer that is used to point to a class method table 1073 in the system director. Based on the information from the file 1061, deputy multimedia device object forming means 1047 generates an object 1068 named "deputy multimedia device." To coordinate inputting and outputting of data between the objects, there is an data I/O coordinating means 343. As there are installed various items of application, their objects are produced by application object forming means 380. Further included are communicating means 1074 for messages, processing and retrieving means 342, a portion 1075 for methods and a portion 1076 for internal data. The internal data are an object ID 1077, management data 344 with which information is formed to link two multimedia devices when these devices are to perform a certain operation, registration data 1078 for information about which of the multimedia devices are in operative connection and what objects are formed.

(0046)

When a multimedia device 2 is connected to the LAN 4, the system director object 205 operates as follows: Using the deputy multimedia device object forming means 1047, the file 1061 for describing the deputy multimedia device object is read in. From the information

obtained from this file 1061, the one of the classes of the general class library 1081 to which the object to be formed should belong is then selected. Based on the instruction in the selected class at the class definition part 1080, the deputy multimedia device object 1068 is formed.

(0047)

Fig. 12 is a diagram of the context of that section of the file for describing a deputy object which describes a control panel. In Fig. 12, the section 247 for describing a control panel object is comprised of a first database 248 for describing the object to an i-th database 249 for descring the object, totaling i databases for describing the object. Each database for describing the object comprises data 250 for recognizing the object. data 254 for drawing the object and data 260 for object linkage.

(0048)

The data 250 for recognizing the object are comprised of a class name 251, as the object belongs to this class, an object ID 252, or distinct ID to the i-th object and an ID of the object to which the i-th object is directly subordinate, or superior object ID 253.

(0049)

The data 254 for drawing the object are used to depict a button or like object constituting part of the control panel window 231 on the screen, being comprised of a first data 255 for drawing the object to

a j-th data 259 for drawing the object, totaling j data for drawing the object. Each data for drawing the object is comprised of data 256 for the location and size at and to which to depict the object, data 257 for the pattern and color and an object image 258.

(0050)

The data 261 for object link are the description that provides information about a link of the controller object 207 or another one of the objects constituting the control panel to a relational object, comprising a first data 261 for object link to a k-th data 264 for object link, totaling k data for object link. One data for object link is comprised of a relational object ID 262 and an message 263 for transmission to the relational object.

(0051)

Fig. 13 is a diagram of the structure of a section for describing the deputy data I/O object in the file for describing the deputy object. In Fig. 13, the section 650 for describing the deputy data I/O object includes a first data 651 for the deputy input object to an m-th data 652 for the deputy input object. Each data for the deputy input object is comprised of an object ID for itself, an ID of that data input object which is selected to link to, or link terminal data input object ID 653 and a list of types of files available to input, or compatible file type list 654. The file section 650 further includes a first data 659 for the deputy

output object to an n-th data 663 for the deputy output object. Each data for the deputy output object is comprised of an object ID 660 of itself, an ID of that object which is selected to output data to, or relational data output object ID 661, and a list of types of files available to output or a compatible file type list 662.

(0052)

Next, the operation of an apparatus employing the above-described system control method of the invention is described by taking a specific example of the multimedia device 2, in this instance, a digital VTR.

(0053)

Fig. 14 is a diagram of the state of the apparatus before an object embedded digital VTR is connected to the multimedia controller. In Fig. 14, 203 is the digital VTR. A digital VTR object 206 always resides in the digital VTR 203 and functions as the object embedded digital VTR as viewed from the other devices on the LAN. The digital VTR object 206 is further constructed from three objects, of which the object 207 controls the hardware of the digital VTR 203, named "digital VTR controller".

(0054)

The second object 208 serves to input digital data as transmitted from another device over the LAN 4, being named "data input to digital VTR". The third object 209 serves to output digital data to another device by transmission over the LAN 4, being named "data output

from digital VTR". When the digital VTR 203 is connected to the multimedia controller 1 via the LAN 4, a deputy digital VTR object is generated in the interior of the multimedia controller 1 based on the specification in a file 210 for describing the deputy digital VTR object.

(0055)

This file comprises a section 211 which describes the specification of a control panel for the digital VTR 203, or section for describing the "control panel for the digital VTR" object and another section 212 which describes the specification of an object that acts as substitute for the data input/output of the VTR 203, or section for describing the "deputy data I/O of digital VTR" object.

(0056)

Fig. 15 is a diagram of the structure of the VTR controller object 207. In this figure, a portion 1009 accommodates a pointer that goes to a class method table 1018. This table is formed with a wide variety of data processing means including means 1019 for activating the play mode to operate under the control of the hardware of the VTR, or play executing means, and means 1020 for activating the record mode to operate or record executing means. 1010 is message communicating means. 1011 is processing and retrieving means. Although a portion for methods is shown at 1012, the actual data processing means are presented by a class method table 1018. The internal data in a portion 1015 are

comprised of an object ID 204, the tape running speed 1016, the current tape footage 1017 and many other variables and status data necessary to control the digital VTR 203.

(0057)

Let us first explain the operation which occurs when the digital VTR 203 is connected to the LAN 4. Fig. 16 is a flowchart of the routine for the operation of the digital VTR when connected to the LAN 4. Fig. 17 is a plan view of a window for the multimedia controller 1 on the screen. In Fig. 17, the multimedia controller window 228 contains a number of icons of which the icon 229 appears when the digital VTR 203 is connected to the LAN 4. Using a mouse or like pointing device, one indicates a location with a cursor 230. The pointing device, though not shown, is provided with buttons. The user presses the button and then releases it. This pressing is usually called clicking. Pressing it twice in a predetermined short interval is called double clicking. Incidentally, as other usable devices, mention may be made of a camera (for inputting still pictures), a tuner, a television set, various relational databases and a CD. To allow the user to select these options, the window 228 displays their icons.

(0058)

Fig. 18 is a diagram to explain the state of the system when the object embedded digital VTR 203 as an example of the multimedia device is connected to the

LAN 4. In Fig. 18, an object 220 is now formed in the interior of the multimedia controller 1. This object functions as substitute for the digital VTR 203, being named "deputy digital VTR". The deputy digital VTR object 220 is constructed from an object 221 which functions as the control panel for the digital VTR 203, named "deputy control panel for digital VTR", another object 222 which, when inputting data, functions as substitute for the data input object 208, named "deputy data input to digital VTR", and yet another object 223 which functions also as substitute for the data output object 209, named "deputy digital output from digital VTR".

(0059)

With reference to Fig. 16, Fig. 17 and Fig. 18, assuming that the object embedded digital VTR 203 as an example of the multimedia device is connected to the LAN 4, the operation of the system is described below. When the digital VTR 203 is connected to the LAN (636), the system director object 205 recognizes the occurrence of connection of the digital VTR 203 (637). Then it sends a device ID to the digital VTR 203 (638).

(0060)

Using the deputy multimedia device object forming means 1047, the system director object 205 then loads the file 210 for describing the deputy digital VTR object from the digital VTR 203 (639). Using the deputy multimedia device object forming means 1047 and based

on the file 210 for describing the deputy digital VTR object, the system director object 205 then generates a deputy digital VTR object 220 in the interior of the multimedia controller 1 (640). Such procedure results in the connected state of the system shown in Fig. 18. Then, the deputy digital VTR object 220 presents the display of an icon 229 of the digital VTR 203 in the multimedia controller window 228 (641). After this, the system stands by for instructions from the user (642).

(0061)

Subsequently, the user will double click the mouse button with the cursor 230 on this icon 229, thereby opening a control panel window based on the digital VTR control panel object 221 of the multimedia controller 1. With the help of the control panel in the windows screen, the user manipulates the digital VTR. So the system can control the digital VTR through the intermediary of the deputy digital VTR object 220 in the multimedia controller 1.

(0062)

Next, the relationship between the content of the file 210 for describing the deputy digital VTR object and the objects to be generated is described in detail below.

(0063)

Fig. 19 is a plan view of an icon of the VTR 203 and Fig. 20 is a plan view of an example of the control panel window on the screen. When the digital VTR 203

is connected to the LAN 4, the icon 229 of Fig. 19 is brought into the device option menu. As the user selects this icon 229, the digital VTR control panel object 221 depicts a window of Fig. 20 by default on the screen. In the same figure, a window option menu 232 is used to selectively display the control panel windows on the screen. As the tape is running, the passed time is displayed in a time counter box 265. As the digital VTR 203 has a number of control modes, there is a control mode option box 267 containing a first switch button for setting a control mode by default and a second switch button 268 for selectively setting more elaborate control modes. A rewind button 269, a reverse play button 270, a pause button 271, a play button 272, a fast feed button 273, a stop button 274 and a record button 275 are displayed in array.

(0064)

Fig. 21 is a diagram, partly in pictorial form, to explain the correspondence between the classes the objects belong to and the constituent elements of the object of the control panel of the digital VTR. All the fundamental constituent elements belong to classes which are defined previously as available in a class library 1081. This library is kept in the multimedia controller 1. As is obvious from Fig. 21, the items in the control panel of the digital VTR, when selected, function as the respective individual objects constituting the digital VTR control panel object 221.

(0065)

Referring to Fig. 21, the frame of a control panel window 231 on the screen corresponds to the VTR control panel object 221 (ID = 1) in a panel class. The item option menu 232 of the control panel window corresponds to the panel view as selectively set from the menu object 285 (ID = 2) in a menu class. For the time counter box 265 to display, the time counter object 286 (ID = 3) in a form class works. For the rewind button 269 to display, the rewind button object 287 (ID = 4) in a button class works. For the reverse play button 270 to display, the reverse play button object 288 (ID = 5) in the button class works. For the pause button 271 to display, the pause button object 289 (ID = 6) in the button class works. For the play button 272 to display, the play button object 290 (ID = 7) in the button class works. For the fast feed button 273 to display, the fast feed button object 291 (ID = 8) in the button class works. For the stop button 274 to display, the stop button object 292 (ID = 9) in the button class works. For the record button 275 to display, the record button object 293 (ID = 10) works.

(0066)

The control mode dialog box 266 corresponds to the control mode selection object 294 (ID = 11). The first switch button 267 corresponds to the default button object 295 (ID = 12) in a radio button class. The second switch button 268 corresponds to the advanced button

object 269 (ID = 13) in the radio button class.

(0067)

Next, of the items constituting the digital VTR control panel object 221 shown in Fig. 21, the play button is taken as an example to explain how to generate an object therefor.

(0068)

Fig. 22 is a diagram of the related parts of the system to the play button object 290. The file 210 for describing the deputy digital VTR object contains a section 247 for describing an object for the control panel of items. This section in turn contains elements 297, 298, 299, 300, 601, 602, 604, 605, 606, 607, 608, 609, 610 and 611.

(0069)

Data 297 for recognizing the object are comprised of a class name 298, an object ID 299 and a superior object ID 300. First data 601 for drawing the object are comprised of a data 602 for location and size, a data 603 for pattern and color and an object image 604. Second data 605 for drawing the object are comprised of a data 606 for location and size, a data 607 for pattern and color and an object image 608. Object link data 609 are comprised of a link terminal object ID 610 and a message 611 to transmit.

(0070)

The play button object 290 is generated by information from the class and also from the section 247

for describing the control panel object in the file 210 for describing the deputy digital VTR object. A portion 613 accommodates a pointer that goes to a class method table 625, where it points to methods in the button class. The button class method table 625 is formed from button initializing means 626 operating in such a manner that when to generate an object in the button class, the internal variables of the button object are initialized, button drawing means 627 for presenting the display of a button drawn from the button object, and click response means 628. The user moves the mouse or like pointing device to position the cursor 230 on top of the drawn play button. Responsive to clicking on this button, the click response means 628 changes the display of the button for a moment to inform the user of the fact that the button object has been activated and sends a message to another object.

(0071)

The definition of every data processing means in the method table for these button classes is described in each class. Therefore, not only the play button object 290 but also all the other objects which belong to the button class share the common button class method table. 614 is message communicating means; 615 is processing and retrieving means; 616 is a portion for methods; and 620 is a portion for internal data. The internal data are an object ID 621, the state of the button, drawing parameters 623 and link data 624. The

one of the types of the internal data which not only the play button object 290 but also all the other button objects which belong to the button class should possess is described in the class.

(0072)

The system director object 205 first reads in the file 210 for describing the deputy digital VTR object, when any of the objects is to generate. In the example of Fig. 22, it then accesses the data for recognizing the object and, on the basis of the description of its class name 298, generates the objects of the button class. For the play button object 290 to generate, the system director object 205 cooperates with the button initializing means 626 to initialize the internal data in the portion 620. According to the example of Fig. 22, the object ID is set to ID = 7 by the description of the object ID 299. From the description of the superior object ID 300, the system director object 205 recognizes that the play button object 290 belongs to the "control panel for digital VTR" object 221. In such a manner, based on the information from the superior object to any of the objects of principal interest, the system director object 205 recognizes which one of the objects contains the other. Thus, a number of the constituent objects are put together to form a complex object.

(0073)

The button drawing means 627 depicts the play

button object 290 on the basis of the drawing parameters 623 and the data 622 of the state of the button. The button drawing means 627 is automatically activated when the button object is generated and when the superior object moves.

(0074)

The first data 601 for drawing the object give off information about a button 625 in the situation when it is not pushed. The data 602 for location and size give off information about a location at which the play button object 290 is depicted in the window for the "control panel for digital VTR" object 221 and about a rectangular frame of which the area determines the size of the play button 625. To figure the rectangular frame, as the button is not pressed, the drawing data for the button 625 are expressed in the coordinates of the "control panel for digital VTR" object 221. Suppose, for example, the left hand upper and right hand lower corners are taken into account, then the data of that rectangular frame have a form like (X1, Y1) and (X2, Y2). For the play button object, when not pressed, its pattern and color are determined based on information either from the pattern and color data 603 or from the object image 604. The data 603 for pattern and color are described in an appropriate language to draw lines and paint colors, that is, to describe the object in a graphical form. The object image 604 is expressed in the form of bit map data. In general, the expression by the former costs

a less amount of data, but the latter has rather a high degree of freedom.

(0075)

The second data 605 for drawing the object drawing, similarly to the first data 601 for drawing the object, describe information about a button 626 in the situation when it is pressed. Based on both of the first data 601 and the second data 605, the values of the drawing parameters 623 are determined. The link data 624 are set in based on information from the data 609 for object linkage. Therefore, as the message to transmit, "play" is set in and, as the link terminal object ID, a link terminal object ID is set in. In connection with the latter, it is to be noted that when to transmit the message, only one of the terminal objects over the entirety of the system should be selected to receive this message. To this purpose, the link terminal object ID to be used is set in the preceded form by the device ID the system director object 205 has assigned to the digital VTR, when the digital VTR 203 was connected to the LAN 4.

(0076)

Even if it happens that two devices have their link terminal objects to use the same ID, therefore, the message can be transmitted right to the desired object. The button state data holds the information of whether or not the button is pressed.

(0077)

Fig. 23 is two flowcharts, one of which shows the operation when the user positions the cursor 229 on top of the icon 229 of the digital VTR 203 and double clicks on it, and another one which shows the operation when the user has manipulated the control panel.

(0078)

Fig. 24 is a plan view of the display of a window for the multimedia controller 1 on the screen as presented when the user has double clicked on the icon 229 of the digital VTR. In Fig. 24, the control panel window 231 for the VTR 203 is selected by default, and the play button is shown at 272.

(0079)

Fig. 29 is a diagram of the relationship between the structure of the "control panel for digital VTR" object in the panel class and the object description data.

(0080)

In Fig. 29, a portion 1401 accommodates a pointer that goes to the class method table, in this instance, a panel class method table 1402. This table is formed with panel initializing means 1403 for initializing the panel object, panel drawing means 1404 for showing the panel in a graphical form, and click response means 1405 for activating the clicked object. Message communicating means 1406, processing and retrieving means 1407, and a portion 1410 for internal data are shown. The internal data are an object ID 1411, a panel state data 1412, and drawing parameters 1413.

The internal data portion 1410 is initialized based on the information from the file 210 for describing the deputy digital VTR object. This file contains a section 211 for describing the "control panel for digital VTR" object, of which the content comprises data 1414 for recognizing the object, first data 1418 for drawing an icon 1426 of the digital VTR 203, and second data 1422 for drawing the frame of the control panel window for the digital VTR. The data 1414 for recognizing the object are comprised of a class name 1415 (panel class), an object ID 1416 (ID = 1), and a superior object ID 2417. The first data 1418 for drawing the object are comprised of data 1419 for location and size, data 1420 for pattern and color, and an object image 1421. The second data 1422 for drawing the object are comprised of data 1423 for location and size, data 1424 for pattern and color and an object image 1425.

(0081)

Referring to Fig. 23 and Fig. 29, the routines for displaying the control panel window for the digital VTR 203 and activating the play mode are described below. As described in connection with the routine of Fig. 16, the system director object 205 generates the deputy digital VTR object 220 in the step 640. At a time when this operation has completed, the deputy digital VTR object 220 presents the display of the icon 229 as obtained based on the icon image 1426. For now, the user double clicks on the icon 229 of the digital VTR (643).

Responsive to this, the control panel object 221 of the deputy digital VTR object 22 sends a message of executing the drawing function to all objects constituting the control panel object 221. Based on this message, all the objects shown in Fig. 21 activate the drawing means. Meanwhile, the control panel object 221 depicts the frame of the control panel window for the digital VTR based on the second object drawing data. As a result, the digital VTR control panel window 231 is displayed on the screen (644) as shown in Fig. 24 and waits for instructions from the user (645). With this, when the user positions the cursor 230 on top of the play button 272 and clicks on it (646), the control panel object 221 sends a message "PLAY" to the controller object 214 of the digital VTR 203 (647). Responsive to this message, the controller object 214 of the digital VTR 203 activates the play executing means (648), thus starting an operation of the play mode of the digital VTR 203.

(0082)

As has been described above, according to the invention, all what has to do for integration of a multimedia device is only connect it to the multimedia controller via the LAN. On connection, its object necessary to coordinate the multimedia device with the others, or deputy multimedia device object, is automatically generated in the multimedia controller. Further, the control panel necessary to choose the

multimedia devices is automatically displayed in the multimedia controller window on the screen. With the help of this control panel, the user activates an item. Then an unique message is transmitted to the controller object of the corresponding multimedia device. So the desired functions are executed. Since the information necessary to generate the deputy multimedia object, which in turn is necessary to manipulate the multimedia device, is obtained from the deputy multimedia device object description file read from the multimedia device, what suffices for the multimedia controller is only the fundamental class library. So there is no need to store the related database to any specific multimedia device in advance.

(0083)

Fig. 25 is a diagram of the relationship between the structure of the "deputy data input to digital VTR" object and the data for describing the object. In Fig. 25, the "deputy data input to digital VTR" object 222 contains a portion 668 for accommodating a pointer that goes to the class method table, in this instance, a deputy data input class method table 679. This table is formed with means 680 for initializing the deputy data input object, means 681 for updating the link data and compatible file type reply means 678.

(0084)

669 is message communicating means; 670 is processing and retrieving means; 671 is a portion for

methods; and 674 is a portion for internal data. The internal data are an object ID 675, another ID 676 which represents the concurrent data input object, or concurrent data input object ID, file types 677 of available data for inputting, or compatible file types, and link data 1006 for linkage with data output objects.

(0085)

The deputy digital VTR object description file 210 contains a "deputy data input/output of digital VTR" object description section 212. Based on this section, the "deputy data input to digital VTR" object is generated. The data for the "deputy input" object described in the section 212 are comprised of an object ID 683 (in this instance, ID = 1), a concurrent data input object ID (in this instance ID = 1), and a list of compatible file types 685 (in this instance, assumed to be formats so called "AV1" and "AV2"). According to the description of these parameters, the deputy input object initializing means 680 initializes the data of the internal data portion 674.

(0086)

Fig. 26 is a diagram of the relationship between the structure of the "deputy data output from digital VTR" object and the data for describing the object. In Fig. 26, the "deputy data output from digital VTR" object 223 contains a portion 690 for accommodating a pointer that goes to the class method table, in this instance, a deputy data output class method table 1048. This table

is formed with means 694 for initializing the deputy data output object, means 695 for updating the link data and compatible file type reply means 700.

(0087)

691 is message communicating means; 692 is processing and retrieving means; 693 is a portion for methods; and 696 is a portion for internal data. The internal data are an object ID 697, another ID 698 which represents the concurrent data output object, or concurrent data output object ID, file types 699 of data available for outputting, or compatible field type, and data 688 for linkage with the data output object.

(0088)

The file 210 for describing the deputy digital VTR contains a section for describing the "deputy data output from digital VTR" object. Based on this section, the "deputy data output from digital VTR" object is generated. The data 1001 for the deputy data output object described in the section 212 are comprised of an object ID 1002 (in this instance, ID = 1), a concurrent data output object ID 1003 (in this instance ID = 1) and a list 1004 of compatible file types (in this instance, assumed to be formats so called "AV1" and "AV2"). According to the description of these parameters, the deputy data output object initializing means initializes the data of the internal data portion 696.

(0089)

Fig. 27 is a diagram of the structure of the

"data input to digital VTR" object. This object 208 includes a portion 1030 for accommodating a pointer that goes to the class method table, in this instance, a data input class method table 1031. This table 1031 is formed with file writing means 1032, data receiving means 1033 and link data updating means 686. 1023 is message communicating means; 1024 is processing and retrieving means; 1025 is a portion for methods; 1028 is a portion for internal data; 1029 is an object ID; and 1030 is link data.

(0090)

Fig. 28 is a diagram of the structure of the "data output from digital VTR" object. This object 209 contains a portion 1035 for accommodating a pointer that goes to the class method table, in this instance, a data output class method table 1044. This table is formed with file reading means 1045, data transmitting means 1046, and link data updating means 687. 1036 is message communicating means; 1037 is processing and retrieving means; 1038 is a portion for methods; 1041 is a portion for internal data; 1042 is an object ID; and 1043 is link data.

(0091)

The digital VTR 103 thus puts its deputy data input object 222 and its deputy data output object 223 into the multimedia controller. After this, these objects function as if they were chief ones, or the "data input to digital VTR" object 208 and the "data output from

digital VTR" object 209. Now suppose the digital VTR receives, for example, a file from another multimedia device by the copy function, then the system director object 205 inquires of the deputy "data input to digital VTR" object 222 what types of files are possible to input. Responsive to this inquiry, the compatible file type reply means of the deputy "data input to digital VTR" object 222 gives off information about the file types the digital VTR 203 can accept.

(0092)

If the type of the file to be copied is found to be present among them, linkage is set in from the deputy output object of that multimedia device which has the file to be copied to the deputy "data input to digital VTR" object 222. The link data updating means 681 of this object 222 sends a message to the "data input to digital VTR" object 208. As the link updating means 686 of this object 208 is activated, the link data of the "data input to digital VTR" object 208 are updated.

(0093)

At the same time, the deputy data output object of that multimedia device which has the file to be copied sends a message for updating the link data of the data output object. As the link data are updated, linkage is set in from the data output object of the multimedia device which has the file to be copied to the "data input to digital VTR" object 208.

(0094)

After this, the data transmitting means of the data output object of the multimedia device which has the file to be copied. The data output object of the multimedia device which has the file to be copied sends a message to the "data input to digital VTR" object. As the data receiving means 1033 and the file writing means 1032 are activated, copying of the file is carried out. In short, when the user gives the copy command or any of the other commands to the deputy data input object and the deputy data output object in the multimedia controller, the deputy data input object and the deputy data output object send the messages to the data input object and the data output object of the main systems of the respective multimedia devices. Thus a linkage for data communication is established between these two multimedia devices. Concerning the operation of copying data, for example, it is not virtually necessary for the multimedia controller to take direct participation.

(0095)

(Advantages of the Invention)

As is understandable from the foregoing, when to make control of the entirety of a system having a plurality of multimedia devices connected to one another, it has been the common practice in the prior art that the device drivers or like applications for this purpose are previously installed in the controller. According to the invention, however, such a necessity is obviated. So all what is needed to do is only connect the multimedia

devices to the LAN. With this, the control panel and the states of the devices are automatically displayed in the controller window on the screen, thereby giving a great advantage that the user can perform any activations in a visual or graphical form, thus quickly and easily turning on and off the electric power sources of the devices, controlling the main systems of the devices, and coordinating inputting and outputting of the various signals and data.

(0096)

Another advantage is that of the items the multimedia devices have sent to the control panel of the controller, the ones which are identical in definition to the items the controller has already possessed may be exchanged either in part or all therebetween, depending on the user's taste. As the user interface differs with different makers, it is thus made possible to unitarily rearrange the items. Yet another advantage is that it becomes possible to execute the control functions from a controller in the distant place and the functions of accessing to the terminal multimedia devices in a transparent fashion over the LAN.

(Brief Description of the Drawings)

(Fig. 1)

A diagram showing a scheme of logical connections of the multimedia controller with the multimedia devices.

(Fig. 2)

Diagrams of physical connections of the multimedia controller to the multimedia devices.

(Fig. 3)

A diagram of the structure of the internal parts of an object embedded multimedia device.

(Fig. 4)

A diagram of the structure of the internal parts of an object embedded multimedia controller.

(Fig. 5)

A diagram of the hierarchy in a system of the multimedia controller.

(Fig. 6)

A diagram of the hierarchy in a system of the multimedia device.

(Fig. 7)

A diagram of the state of the system before the multimedia device is connected to the multimedia controller.

(Fig. 8)

A diagram of the state of the system after the multimedia device is connected to the LAN.

(Fig. 9)

A diagram of the structure of a general class library.

(Fig. 10)

A diagram of the structure of an object.

(Fig. 11)

A diagram of the structure of a system director object.

(Fig. 12)

A diagram of the structure of a section for describing the control panel in the file for describing the deputy object.

(Fig. 13)

A diagram of the structure of a section for describing the data input/output object in the file for describing the deputy object.

(Fig. 14)

A diagram of the state of the system before an object embedded digital VTR is connected to the multimedia controller.

(Fig. 15)

A diagram of the structure of the digital VTR controller object.

(Fig. 16)

A flowchart for the operation when the digital VTR is connected to the LAN.

(Fig. 17)

A plan view of a multimedia controller window on the screen.

(Fig. 18)

A diagram of the state of the system when the object embedded digital VTR is connected as the multimedia device to the LAN.

(Fig. 19)

A plan view of an icon of the digital VTR.

(Fig. 20)

A plan view of a control panel window on the screen.

(Fig. 21)

A diagram of the correspondence between the classes to which the objects belong and the constituent elements of the digital VTR control panel object.

(Fig. 22)

A diagram to explain the generation of a play button object.

(Fig. 23)

Flowcharts for the operations when the user double clicks on the icon of the digital VTR in the control panel window and then on the play button in the digital VTR controller window.

(Fig. 24)

A plan view of a multimedia controller window after the user has clicked on the icon of the digital VTR in the control panel window.

(Fig. 25)

A diagram of the relationship between the structure of the deputy "data input to digital VTR" object and the object description data.

(Fig. 26)

A diagram of the relationship between the structure of the deputy "data output from digital VTR" object and the data for describing the object.

(Fig. 27)

A diagram of the structure of the "data input to digital VTR" object.

(Fig. 28)

A diagram of the structure of the "data output from digital VTR" object.

(Fig. 29)

A diagram of the relationship between the structure of the digital VTR control panel object of the panel class and the object description data.

24.04.95

(Name of Document)

Drawings

(Fig. 1)

Logical Forms of Connection of Multimedia Controller with multimedia devices

1: Multimedia Controller

2: Multimedia Device (Digital Camera)
2: Multimedia Device (Printer)
2: Multimedia Device (Digital FAX)
2: Multimedia Device (Digital Copier)
2: Multimedia Device (Digital VTR)
2: Multimedia Device (CD Player)

(Fig. 2)

Physical Forms of connection of Multimedia Controller with Multimedia Devices

a) Daisy Chain Line
   1: Multimedia Controller
   2: Multimedia Device

b) Star Configuration
   1: Multimedia Controller
   2: Multimedia Device

c) Multipoint Line
   1: Multimedia Controller
   2: Multimedia Device

(Fig. 3)

Internal Structure of Object Embedded Multimedia Device

10: Internal Bus
15: Multimedia Data
16: Driver for Mechanical System
17: Mechanisms & Motors
18: Driver for Electrical System
19: Electrical Circuits, Indicators & Switches

(Fig. 4)

Internal Structure of Object Embedded Multimedia Controller

25: Multimedia Data Filing
26: Display Controller
27 Display
28: Driver for Electrical System
29: Electrical Circuits, Indicators & Switches

(Fig. 5)

System Hierarchy of Multimedia Controller

50: Hardware
53: Common Class Library
54: C Function
55: Specific Class Library
56: Application for Controller for Multimedia Devices

(Fig. 6)

System Hierarchy of Multimedia Device

57: Hardware
59: Specific Class Library
60: C Function
61: Application for Multimedia Device

(Fig. 7)

1: Multimedia Controller
    205: System Director Object

2: Multimedia Device
    1061: File for Describing Deputy Multimedia Device Object
        1062: Section for Describing Multimedia Device Control Panel Object
        1063: Section for Describing Deputy "Data I/O of Multimedia Device" Object
    1064: Multimedia Device Object
        1065: Multimedia Device Controller Object
        1066: "Data Input to Multimedia Device" Object
        1067: "Data Output from Multimedia Device" Object

(Fig. 8)

1: Multimedia Controller
- 205: System Director Object
- 1068: Deputy Multimedia Device Object
  - 1069: Multimedia Device Control Panel Object
  - 1070: Deputy "Data Input to Multimedia device" Object
  - 1071: Deputy "Data Output from Multimedia Device" Object

2: Multimedia Device
- 1064: Multimedia Device Object
  - 1065: Multimedia Device Controller Object
  - 1066: "Data Input to Multimedia Device" Object
  - 1067: "Data Output from Multimedia Device" Object

(Fig. 9)

1081: Class Library
- 1079: First Class
  - 1080: Class Definition Part
  - 1081: Class Method Table
  - 1082: Code Part
    - 1083: 1st Function Code
    - 1084: k-th Function Code
- 1085: p-th Class

(Fig. 10)

234: Object
- 235: Portion for Internal Data
  - 236: 1st Internal Data
  - 237: 2nd Internal Data
  - 238: n-th Internal Data
- 239: Portion for Methods
  - 240: 1st Data Processing Means
  - 241: 2nd Data Processing Means
  - 242: m-th Data Processing Means
- 243: Class Method Table
  - 240: 1st Data Processing Means
  - 241: 2nd Data Processing Means
  - 242: 3rd Data Processing Means
- 244: Portion for Accommodating Pointer for Class Method Table
- 245: Message Communicating Means; Message
- 246: Processing & Retrieving Means

External Data

(Fig. 11)

205: System Director Object
- 342: Processing & Retrieving Means
- 1061: File for Describing Deputy Multimedia Device Object
- 1072: Portion for Accommodating Pointer for Class Method Table
- 1073: System Director Class Method Table
  - 1047: Deputy Multimedia Device Object Forming means
  - 343: Data I/O Coordinating Means
  - 380: Application Object Forming Means
- 1074: Message Communicating Means; Message
- 1075: Portion for Methods
  - 1047: Deputy Multimedia Device Object Forming means
  - 343: Data I/O Coordinating Means
  - 380: Application Object Forming Means
- 1076: Portion for Internal Data
  - 1077: Object ID
  - 1078: Object Registration Data
  - 344: Management Data for link between Devices

(Fig. 12)

247: File Section for Describing Control Panel Object
- 248: 1st Database for Describing Object
  - 250: Data for Recognizing Object
    - 251: Class Name
    - 252: Object ID
    - 253: Super Object ID
  - 254: Data for Drawing Object
    - 255: 1st Data for Drawing Object
      - 256: Data for Location & Size
      - 257: Data for Pattern & Color
      - 258: Object Image
    - 259: j-th Data for Drawing Object
  - 260: Data for Object Link
    - 261: 1st Data for Object Link
      - 262: Relational Object ID
      - 263: Message for Transmitting
    - 264: k-th Data for Object Link
- 249: i-th Database for Describing Object

(Fig. 13)

650: File for Describing Deputy Data I/O Object
- 651: 1st Data for Deputy Data Input Object
  - 652: Object ID
  - 653: Relational Data Input Object ID
  - 654: Compatible File Type List

655: m-th Data for Deputy Data Input Object
659: 1st Data for Deputy Data Output Object
660: Object ID
661: Relational Data Output Object ID
662: Compatible File Type List
663: n-th Data for Deputy Data Output Object

(Fig. 14)

1: Multimedia Controller
205: System Director Object

203: Digital VTR
206: Digital VTR Object
207: Digital VTR Controller Object
208: "Data Input to Digital VTR" Object
209: "Data Output from Digital VTR" Object

210: File for Describing Deputy Digital Control Panel Object
211: Section for Describing Digital Control Panel Object
212: Section for Describing Deputy "Data Output from Digital VTR" Object

(Fig. 15)

207: Digital VTR Controller Object
1009: Portion for Accommodating Pointer for Class Method Table
1010: Message Communicating Means
1011: Processing & Retrieving Means
1012: Portion for Methods
1019: Play Executing Means
1020: Recording Executing Means
1015: Portion for Internal Data
204: Object ID
1016: Tape Running State
1017: Current Footage
1018: Controller Class Method Table
1019: Play Executing Means
1020: Recording Executing Means

(Fig. 16)

636: Connect Digital VTR to Network
637: System Director Object Checks When Digital VTR is Connected
638: System Director Object Gives Digital VTR Device ID
639: System Director Object Loads Deputy Digital VTR Object Description File

640: System Director Object Generates Deputy Digital VTR Object Based On Information from File for Describing Deputy Digital VTR Object
641: Deputy Digital VTR Object Displays Icon (Mini Panel) of Digital VTR in System Controller Window
642: Wait for Instruction from User

(Fig. 17)

(Fig. 18)

1: Multimedia Controller
- 205: System Director Object
- 220: Deputy Digital VTR Object
  - 221: Deputy "Control Panel for Digital VTR" Object
  - 222: Deputy "Data Input to Digital VTR" Object
  - 223: Deputy "Data Output from Digital VTR" Object

203: Digital VTR
- 206: Digital VTR Object
  - 207: Digital VTR Controller Object
  - 208: "Data Input to Digital VTR" Object
  - 209: "Data Output from Digital VTR" Object

(Fig. 19)

(Fig. 20)

(Fig. 21)

221: Panel Class: Control Panel for Digital VTR
- 285: Menu Class: Panel View Option Menu
- 286: Form Class: Timer Counter
- 288: Button Class: Rewind Button
- 289: Button Class: Reverse Play Button
- 290: Button Class: Pause Button
- 291: Button Class: Fast Feed Button
- 292: Button Class: Stop Button
- 293: Button Class: Recording Button
- 294: Button Group Class: Selection of Control Modes
  - 295: Radio Button Class: Default Button
  - 296: Radio Button Class: Advanced Button

(Fig. 22)

(Fig. 22)

612: Play Button Object
- 613: Portion for Accommodating Pointer for Class Method Table
- 614: Message Communicating Means
- 615: Processing & Retrieving Means
- 616: Portion for Methods
  - 626: Button Initializing Means
  - 627: Button Drawing Means
  - 628: Click Response Means
- 620: Portion for Internal Data
  - 621: Object ID
  - 622: Data for Button State
  - 623: Drawing Parameter
  - 624: Link Data
- 625: Button Class Method Table
  - 626: Button Initializing Means
  - 627: Button Drawing Means
  - 628: Click Response Means

625: Button When Not Pressed
626: Button When Pressed
297: Data for Object Recognition
- 298: Class Name: Button Class
- 299: Object ID: ID=7 Play Button
- 300: Super Object ID; ID=1 VTR Control Panel

601: 1st Data for Drawing Object
- 602: Data for Location & Size
- 603: Data for Pattern & Color
- 604: Object Image

605: 2nd Data for Drawing Object
- 606: Data for Location & Size
- 607: Data for Pattern & Color
- 608: Object Image

609: Object Link Data
- 610: Link Terminal Object ID ; ID of VTR Control Object
- 6121: Message to Transmit; Play

(Fig. 23)

643: User Double Clicks on Icon of Digital VTR
644: "Control Panel for Digital VTR" Object Presents Display of Control Panel for Digital VTR
645: Wait for User's Action

646: Click on Play button
647: "Control Panel for Digital VTR" Object Sends "Play" Message to Digital VTR Controller Object
648: Digital VTR Controller Object Activates Play Executing Means
649: Start Play Mode of Digital VTR

(Fig. 24)

(Fig. 25)

222: Deputy "Data Input to Digital VTR" Object
- 668: Portion for Accommodating Pointer for Class Method Table
- 669: Message Communicating Means
- 670: Processing & Retrieving Means
- 671: Portion for Methods
  - 680: Means for Initializing Deputy Data Input Object
  - 681: Link Data Updating Means
  - 678: Compatible File Type Replying Means
- 674: Portion for Internal Data
  - 675: Object ID
  - 676: Relational Data Input Object ID
  - 677: Compatible File Types
  - 1006: Link Data
- 679: Deputy Data Input Class Method Table
  - 680: Means for Initializing Deputy Data Input Object
  - 681: Link Data Updating Means
  - 678: Compatible File Type Replying Means
- 682: Deputy Data Input Object Data
  - 683: Object ID
  - 684: Relational Data Input Object ID
  - 685: Compatible File Type List

(Fig. 26)

223: Deputy "Data Output from Digital VTR" Object
- 690: Portion for Accommodating Pointer for Class Method Table
- 691: Message Communicating Means
- 692: Processing & Retrieving Means
- 693: Portion for Methods
  - 694: Means for Initializing Deputy Data Input Object
  - 695: Link Data Updating Means
  - 700: Compatible File Type Replying Means
- 696: Portion for Internal Data
  - 697: Object ID
  - 698: Relational Data Output Object ID
  - 699: Compatible File Types
  - 688: Link Data
- 1048: Deputy Data Output Class Method Table
  - 694: Means for Initializing Deputy Data Output Object
  - 695: Means for Sending Data Input Command
  - 700: Compatible File Type Replying Means
- 1001: Deputy Output Object Data
  - 1002: Object ID
  - 1003: Relational Data Output Object ID
  - 1004: Compatible File Type List

(Fig. 27)

208: "Data Input to Digital VTR" Object
- 1022: Portion for Accommodating Pointer for Class Method Table
- 1023: Message Communicating Means
- 1024: Processing & Retrieving Means
- 1025: Portion for Methods
  - 1032: File Writing Means
  - 1033: Data Receiving Means
  - 686: Link Data updating Means
- 1028: Portion for Internal Data
  - 1029: Object ID
  - 1030: Link Data
- 1031: Data Input Class Method Table
  - 1032: File Writing Means
  - 1033: Data Receiving Means
  - 686: Link Data Updating Means

(Fig. 28)

209: "Data Output from Digital VTR" Object
- 1035: Portion for Accommodating Pointer for Class Method Table
- 1036: Message Communicating Means
- 1037: Processing & Retrieving Means
- 1038: Portion for Internal Data
  - 1045: Data Reading Means
  - 1046: Data Transmitting Means
  - 687: Link Data Updating Means
- 1041: Portion for Internal Data
  - 1042: Object ID
  - 1043: Link Data
- 1044: Data Output Class Method Table
  - 1045: Data Reading Means
  - 1046: Data Transmitting Means
  - 687: Link Data Updating Means

(Fig. 29)

221: "Control Panel for Digital VTR" Object
- 1401: Portion for Accommodating Pointer for Class Method Table
- 1402: Panel Class Method Table
  - 1403: Panel Initializing Means
  - 1404: Panel Drawing Means
  - 1405: Click Response Means
- 1408: Portion for Methods
  - 1403: Panel Initializing Means
  - 1404: Panel Drawing Means
  - 1405: Click Response Means
- 1410: Portion for Internal Data

1411: Object ID
1412: Data for Panel State
1413: Drawing Parameters
1414: Data for Object Recognition
1415: Class Name; Button Class
1416: Object ID; ID = 1 Digital VTR Control Panel
1417: Super Object ID
1418: 1st Data for Drawing Object
1419: Data for Location & Size
1420: Data for Pattern & Color
1421: Object Image
1422: 2nd Data for Drawing Object
1423: Data for Location & Size
1424: Data for Pattern & Color
1425: Object Image
1426: Icon Image
1427: Control Panel for Digital VTR (Frame)

24.04.95

(Name of Document)

Written Abstract

(Abstract)

(Object) To realize an environment that permits any one of the multimedia devices to be controlled without the necessity of special software such as applications or device drivers but with the use of a LAN from the other controllers transparently.

(Constitution) A control system for an apparatus, the apparatus comprising a plurality of object embedded peripheral devices and a controller connected through a common communication line to the plurality of peripheral devices to unitarily control the plurality of peripheral devices, wherein the aforesaid controller operates in such a manner that, when each of the plurality of peripheral devices is connected to the controller, the control data stored in the connected one of the peripheral devices is transmitted over the common communication line to, and written in a predetermined format in, a predetermined memory area in the controller, whereby it is made possible for the controller to control by itself the aforesaid peripheral device, and wherein the commands given by the controller in itself are transmitted over the communication line to each of the peripheral devices.

(Selected Figure) Fig. 1